INFORMATION AND CONTROL 67, 212–222 (1985)

The Critical Complexity of All (Monotone) Boolean Functions and Monotone Graph Properties

INGO WEGENER*

FB 20-Informatik, Johann Wolfgang Goethe-Universität, 6000 Frankfurt a.M., West Germany

CREW–PRAM's are a powerful model of parallel computers. Lower bounds for this model are rather general. Cook, Dwork, and Reischuk upper and lower time bounds for parallel random access machines without simultaneous writes, *SIAM J. Comput.* (in press) proved that the CREW–PRAM complexity of Boolean functions is bounded by $\log_b c(f)$, where $b \approx 4.79$ and $c(f)$ is the critical complexity of f. This lower bound is often even tight. For a class of functions F the critical complexity $c(F)$, the minimum of all $c(f)$ where $f \in F$, is the best general lower bound on the critical complexity of all $f \in F$. We determine the critical complexity of the set of all nondegenerate Boolean functions and all monotone nondegenerate Boolean functions up to a small additive term. And we compute exactly the critical complexity of the class of all monotone graph properties, proving partially a conjecture of Turán (1984).

1. INTRODUCTION

Research on the critical complexity of Boolean functions is motivated by results of Cook and Dwork (1982) and Cook, Dwork, and Reischuk (in press) on the complexity of CREW–PRAM's (concurrent read exclusive write parallel random access machines). For the purpose of this paper it is sufficient to know that CREW–PRAM's are a computation model consisting of an arbitrary number of processors which are RAM's and one common memory tape. Many processors may read the same cell of the common memory tape at the same time but only one processor is allowed at each time to try to write into each cell of the common memory tape otherwise the program is incorrect. For the computation of a Boolean function f: $\{0, 1\}^n \to \{0, 1\}$ the n input bits are given in the first n cells of the common tape and the output must be written in the first cell of this tape. The main result of Cook and Dwork (1982) and Cook, Dwork, and Reischuk (in press) states that the CREW–PRAM complexity of any

* Supported in part by DFG-Grant We 1066/1-1.

0019-9958/85 $3.00

Boolean function f: $\{0, 1\}^n \to \{0, 1\}$ is bounded below by $\log_b c(f)$, where $b = (5 + \sqrt{21})/2 \approx 4.79$ and $c(f)$ is the critical complexity of f. This lower bound plays an important role since the CREW–PRAM model is a general one and since the lower bound is tight up to constant factors for many functions. Now we define the critical complexity of Boolean functions.

DEFINITION 1. Let f: $\{0, 1\}^n \to \{0, 1\}$ be a Boolean function; $c(f, x)$, the number of neighbors y of x (Hamming distance 1), where $f(y) \neq f(x)$, is the critical complexity of f at point x; $c(f)$, the maximum of all $c(f, x)$, where $x \in \{0, 1\}^n$, is the critical complexity of f.

Though we know the critical complexity of many functions, we have no general efficient procedure for the computation of $c(f)$. We are interested in the critical complexity of classes of Boolean functions in order to obtain lower bounds on the critical complexity of the members of these classes.

DEFINITION 2. For a class of Boolean functions F, the minimum of all $c(f)$, where $f \in F$, is the critical complexity $c(F)$ of F.

In this paper we investigate several classes of functions defined below. A Boolean function is called nondegenerate if it depends essentially on all its variables, i.e., if the subfunctions of f for $x_i = 0$ and $x_i = 1$ are different.

DEFINITION 3. (i) B_n is the class of all nondegenerate Boolean functions on n variables.

(ii) M_n is the class of all monotone functions in B_n.

(iii) S_n is the class of all symmetric functions in B_n, i.e., for all permutations $\pi \in \Sigma_n$ on $\{1,..., n\}$ we have $f(x_1,..., x_n) = f(x_{\pi(1)},..., x_{\pi(n)})$.

(iv) G_n is the class of all nonconstant graph properties on graphs with n vertices. G_n contains functions f on $\binom{n}{2}$ variables $x_{i,j}$ $(1 \leqslant i < j \leqslant n)$ corresponding to the possible edges of a graph on n vertices. f is called a graph property if any renumbering of the vertices does not change the value of f, i.e., for all $\pi \in \Sigma_n$

$$f(x_{1,2},..., x_{n-1,n}) = f(x_{\pi(1),\pi(2)},..., x_{\pi(n-1),\pi(n)}).$$

(v) MG_n is the class of all monotone nonconstant graph properties.

In Section 2 we show that the critical complexity of symmetric functions can be computed efficiently. In Section 3 we compute $c(B_n)$ and $c(M_n)$ up to an additive term of small order. Furthermore, we give a characterization of the critical complexity of monotone functions in terms of their prime implicants and prime clauses. Turán (1984) conjectures that $c(G_n) = c(MG_n) = n - 1$. We are able to prove (Sect. 4) that $c(MG_n) = n - 1$. The more general conjecture $c(G_n)=n-1$ remains open. Finally we show in Section 5 that the easiest (monotone) nondegenerate Boolean function and the easiest monotone graph property with respect to the critical complexity belong also to the easiest functions with respect to the complexity of CREW–PRAM's.

2. Symmetric Functions

Already Turán (1984) has proved that $c(S_n)=\lceil (n+1)/2 \rceil$. Here we give an easy procedure to compute efficiently the critical complexity of each $f \in S_n$. The result of Turán is a corollary of this characterization. These considerations are also useful to become familiar with the new complexity measure called critical complexity.

It is well known that symmetric functions f may be described by their value vector $v(f)=(v_0,..., v_n)$ such that $f(x)=v_i$ if x contains exactly i ones.

Proposition 1. *Let $v(f)=(v_0,..., v_n)$ be the value vector of $f \in S_n$. If for some i $(v_{i-1}, v_i, v_{i+1}) \in \{(0, 1, 0), (1, 0, 1)\}$ the critical complexity of f is n. Otherwise* $c(f)=\max\{k+1, n-k \mid v_k \neq v_{k+1}\}$.

Proof. Let x be an input with exactly i ones. Then i neighbors of x have $i-1$ ones and $n-i$ neighbors have $i+1$ ones. If $v_{i-1} \neq v_i \neq v_{i+1}$ input x is n-critical. If $v_{i-1} \neq v_i = v_{i+1}$ input x is i-critical. If $v_{i-1} = v_i \neq v_{i+1}$ input x is $(n-i)$-critical. Finally if $v_{i-1}=v_i=v_{i+1}$ input x is 0-critical. By these observations we have proved the characterization of $c(f)$ for $f \in S_n$. ∎

Corollary 1 (Turán (1984)). $c(S_n)=\lceil (n+1)/2 \rceil$.

Proof. If $f \in S_n$ there exists some k such that $v_k \neq v_{k+1}$. Since $\max\{k+1, n-k\} \geq \lceil (n+1)/2 \rceil$ for all k the lower bound follows. The majority function is defined by $v_k=0$ iff $k<\lceil (n+1)/2 \rceil$. Thus its critical complexity equals $\max\{\lceil (n+1)/2 \rceil, n+1-\lceil (n+1)/2 \rceil\}=\lceil (n+1)/2 \rceil$. ∎

The critical complexity of $f \in S_n$ may be computed by our characterization above in linear time from the value vector $v(f)$.

3. Nondegenerate Boolean Functions and Nondegenerate Monotone Functions

Simon (1983) used counting arguments for the following lower bound.

Theorem 1 (Simon (1983)). $\frac{1}{2}\log n - \frac{1}{2}\log\log n + \frac{1}{2} < c(B_n) \leq c(M_n)$.

We show that these lower bounds on the critical complexity of B_n and M_n are optimal up to additive terms of order $O(\log\log n)$. The previous best upper bound for $c(B_n)$ was a $(\log n+2)$-bound proved also by Simon (1983). Before improving this upper bound we characterize the critical complexity of monotone functions by the lengths of their prime implicants and prime clauses. This characterization turns out to be useful also in the following section.

PROPOSITION 2. *If f is monotone $c(f)$ equals the maximum length of a prime implicant or a prime clause of f.*

Proof. For the lower bound it is sufficient to prove that the existence of a prime implicant or a prime clause of length k implies the existence of a k-critical input. Let $t(x)=x_{i_1}\cdots x_{i_k}$ be a prime implicant of length k and let $a=(a_1,..., a_n)$ be the input where $a_{i_1}=\cdots=a_{i_k}=1$ and $a_j=0$ for all other j. Since $t(a)=1$ also $f(a)=1$. It is well known from the theory of monotone Boolean functions that t is the only prime implicant of f which computes 1 for input a. If we consider a neighbor of a with only $k-1$ ones $t(b)=0$ and by monotonicity also all other prime implicants of f compute 0 implying $f(b)=0$. Thus a is a k-critical input for f. For a prime clause $cl(x)=x_{i_1}\vee\cdots\vee x_{i_k}$ the dual arguments work for the input a', where $a'_{i_1}=\cdots=a'_{i_k}=0$ and $a'_j=1$ for all other j.

For the upper bound we consider an input a such that $f(a)=1$. The case $f(a)=0$ may be handled by similar arguments. Since $f(a)=1$ there exists at least one prime implicant t of f computing 1. For neighbors b of a, where we have switched some 0 to 1 by monotonicity $f(b)=1$. If b is a neighbor of a such that $f(b)=0$ then $b_i=0$ and $a_i=1$ for exactly one i and $a_j=b_j$ for all $j\neq i$. Since $f(b)=0$ all prime implicants including t compute 0 on input b. But $t(a)=1$ and $t(b)=0$ implies that x_i is a variable of t. Thus $c(f, a)$ is bounded above by the length of any prime implicant t such that $t(a)=1$. ∎

Now we present a nearly easiest (monotone) Boolean function with respect to the critical complexity.

DEFINITION 4. Let $N=\binom{n}{\lfloor n/2\rfloor}$ and α be an arbitrary one-to-one mapping from the subsets A of $\{1,..., n\}$ of cardinality $\lfloor n/2\rfloor$ to $\{1,..., N\}$. Let T_k be the kth threshold function computing 1 iff the input contains at least k ones. Finally $f\colon \{0,1\}^{n+N}\to\{0,1\}$ is defined by

$$f(x_1,..., x_n, y_1,..., y_N)$$

$$= T_{\lfloor n/2\rfloor+1}(x_1,..., x_n)\vee\bigvee_{A\subseteq\{1,...,n\},|A|=\lfloor n/2\rfloor}\Big(\bigwedge_{i\in A} x_i\wedge y_{\alpha(A)}\Big).$$

f is a monotone address function. The x vector is the address but only addresses with exactly $\lfloor n/2 \rfloor$ ones are admissible. That means the admissible addresses build an antichain in the lattice $\{0, 1\}^n$. y_i is the content of the storage cell whose address is $\alpha^{-1}(i)$.

PROPOSITION 3. *The critical complexity of the monotone address function* (*of Def.* 4) *is* $\lceil n/2 \rceil + 1$.

Proof. We have defined f in its monotone disjunctive normal form. All prime implicants have length $\lfloor n/2 \rfloor + 1$. By the proof of Proposition 2 all inputs where f computes 1 are at most $(\lfloor n/2 \rfloor + 1)$-critical. We have to consider only inputs where f computes 0. That means that l, the number of ones among the x-variables, is at most $\lfloor n/2 \rfloor$.

Case 1. $l \leqslant \lfloor n/2 \rfloor - 2$. If f computes 1 there must be at least $\lfloor n/2 \rfloor$ ones among the x-variables. Thus these inputs are 0-critical.

Case 2. $l = \lfloor n/2 \rfloor - 1$. By the comment to Case 1 we have to switch an x input from 0 to 1 to obtain an input where f computes 1. Exactly $n - (\lfloor n/2 \rfloor - 1) = \lceil n/2 \rceil + 1$ x variables can be switched from 0 to 1. Thus these inputs are at most $(\lceil n/2 \rceil + 1)$-critical. If for example all y variables are 1 these inputs are indeed $(\lceil n/2 \rceil + 1)$-critical.

Case 3. $l = \lfloor n/2 \rfloor$ and f computes 0. Switching one of the $\lceil n/2 \rceil$ x variables which are 0 changes the value of f. Let A be the set of indices i such that $x_i = 1$. Since $f(x, y) = 0$ we have $y_{\alpha(A)} = 0$. Switching this variable switches also f, but switching one of the other y variables does not change the value of f, since the corresponding address implicant is 0. Thus these inputs are $(\lceil n/2 \rceil + 1)$-critical. ∎

The monotone address function obviously is nondegenerate and monotone. For an upper bound on $c(B_m)$ and $c(M_m)$ we need a sequence of functions $(g_m)_{m \in \mathbb{N}}$ defined on m variables. If $n - 1 + \binom{n-1}{\lfloor (n-1)/2 \rfloor} < m \leqslant n + \binom{n}{\lfloor n/2 \rfloor}$ we define g_m to be equal to the monotone address function on $n + \binom{n}{\lfloor n/2 \rfloor}$ variables, where $(n + \binom{n}{\lfloor n/2 \rfloor} - m)$ y variables are replaced by constants 1. By the same arguments as for the proof of Proposition 3, f_m is at most $(\lceil n/2 \rceil + 1)$-critical. By the formula of Stirling, $m \geqslant n + c 2^{n-1} n^{-1/2}$ for some positive constant c. Thus

$$\log m \geqslant n - \tfrac{1}{2} \log n - \Omega(1),$$

$$\tfrac{1}{2} \log \log m \geqslant \tfrac{1}{2} \log n - \Omega(1),$$

$$n \leqslant \log m + \tfrac{1}{2} \log \log m + O(1),$$

and

$$c(f_m) \leqslant \tfrac{1}{2} \log m + \tfrac{1}{4} \log \log m + O(1).$$

Altogether we have proved

THEOREM 2. $c(B_n) \leqslant c(M_n) \leqslant \frac{1}{2}\log n + \frac{1}{4}\log\log n + O(1)$.

By Theorem 1 and Theorem 2 we know the critical complexity of the classes B_n and M_n up to small additive terms.

4. MONOTONE GRAPH PROPERTIES

THEOREM 3 (Turán (1984)). $\lfloor n/4 \rfloor \leqslant c(G_n) \leqslant n-1$.

Turán conjectures that $c(G_n) = n-1$. His example for the upper bound is the graph property P: "No vertex is isolated." It is a nice exercise to prove that this graph property is $(n-1)$-critical. Since P is obviously a monotone graph property also $c(MG_n) \leqslant n-1$. We are not able to prove the general conjecture of Turán but we prove the simpler conjecture $c(MG_n) = n-1$. This is an interesting subcase since many of the important and interesting graph properties are monotone. The rest of this section is devoted to the proof of $c(MG_n) = n-1$.

THEOREM 4. $c(MG_n) = n-1$.

Proof. As already mentioned, the upper bound was known before. Let P be a monotone graph property for graphs on n vertices. We have to prove $c(P) \geqslant n-1$. By an easy case inspection we prove the claim for $n \leqslant 3$. For $n \geqslant 4$ we use again Proposition 2. Prime implicants correspond to minimal graphs with property P while prime clauses correspond to maximal graphs with property $\bar{P}$, the complementary property of P.

If there exists a minimal graph with P containing at least $n-1$ edges we are done. Thus we may assume

(*) Each minimal graph with property P has at most $n-2$ edges.

Under this assumption we construct a maximal graph with property $\bar{P}$ and at least $n-1$ missing edges. Since the missing edges correspond to a prime clause the Theorem is proved.

In order to explain the idea of the proof and since this subcase is not covered by our general calculations we assume at first that all minimal graphs G^* with P have no isolated vertex. Because of (*) the sum over the degrees of the vertices in G^* is bounded by $2n-4$. Since the degree of each vertex in a minimal graph G^* with P is at least 1 by our assumption, there exists a vertex v in G^* with $\deg(v) = 1$. We consider a complete graph K_{n-1} on $n-1$ vertices together with an isolated vertex u. By our assumption this graph G has $\bar{P}$. If we add one of the $n-1$ missing edges—say (u, w)—the new graph G' has P. By renumbering the vertices we identify u and v and w and the neighbor of v in G^*. Thus all edges of G^* are contained in G' and by monotonicity G' has P. Thus G is $(n-1)$-critical for P.

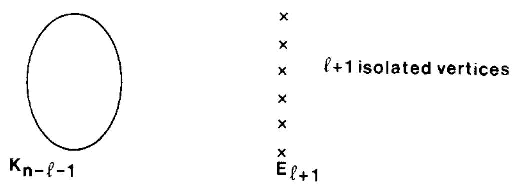

FIG. 1. G' with $\bar{P}$.

In general we have to work harder. Let l be the maximal number of isolated vertices of a minimal graph with property P. By our considerations above we may assume $l>0$. Let $m(G)$ be the minimal degree of a non-isolated vertex in G. Let $m:=\min\{m(G) \mid G$ is minimal with P, G has l isolated vertices$\}$.

Claim. $m \leqslant \lfloor (2n-4)/(n-l) \rfloor$. Let G be a graph defining m. Again by (*) the sum over the degrees of the vertices is bounded by $2n-4$. By definition of m the degree of exactly $n-l$ vertices is at least 1. Thus the smallest degree of a nonisolated vertex is bounded by $(2n-4)/(n-l)$. Since m is an integer the claim is proved.

We now explain the construction of a maximal graph with $\bar{P}$ and at least $n-1$ missing edges. In the first step we consider G', a complete subgraph on $n-l-1$ vertices (K_{n-l-1}) together with $l+1$ isolated vertices (E_{l+1}) (see Fig. 1). By definition of l G' has $\bar{P}$.

Now we add as many edges as possible between the $l+1$ isolated vertices and the $n-l-1$ vertices of K_{n-l-1} such that the new graph G'' has $\bar{P}$ (see Fig. 2). The degree of the former isolated vertices is bounded by $m-1$. If such a vertex v would have degree m or larger the graph would have P. We could identify the other l former isolated vertices and the l isolated vertices of G^*, a minimal graph with P defining l and m. v could be identified with a vertex w of G^* whose degree is m and m neighbors of v could be identified with the neighbors of w. Then all edges of G^* are edges of G'' and by monotonicity G'' has P.

At last we add as many edges as possible among the $l+1$ former isolated vertices such that the new graph G has $\bar{P}$ (see Fig. 3). G is a maximal graph

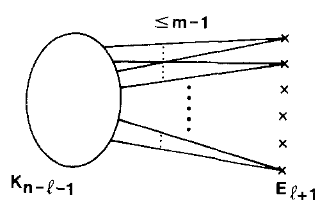

FIG. 2. G'' with $\bar{P}$.

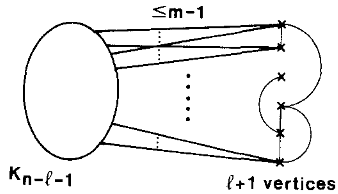

FIG. 3. G, a maximal graph with $\bar{P}$.

with $\bar{P}$. We are done if we can show that there are at least $n-1$ missing edges in G.

At first we estimate the number of edges missing between K_{n-l-1} and the other $l+1$ vertices. Since at most $m-1$ of the possible $n-l-1$ vertices K_{n-l-1} are connected with a definite of the $l+1$ vertices, at least $(n-l-1-(m-1))(l+1)$ edges are missing. We are done if $(n-l-m)(l+1)\geqslant n-1$. By our claim above it is sufficient that $(n-l-(2n-4)/(n-l))(l+1)\geqslant n-1$. Since $n-l>0$ this is equivalent to $l^3-2l^2n+l^2+ln^2-3ln+3l\geqslant n-4$.

Since $l\geqslant 1$ it is sufficient that

$$l^2-2ln+l+n^2-3n+3\geqslant n-4$$

$$\Leftrightarrow l^2-2l\left(n-\frac{1}{2}\right)+\left(n-\frac{1}{2}\right)^2\geqslant 3n-\frac{27}{4}.$$

Therefore it is sufficient that

$$l\leqslant n-\tfrac{1}{2}-\sqrt{3n-27/4}.$$

We distinguish two cases:

Case 1. K_{n-l}, the complete subgraph on $n-l$ vertices, is not the minimal graph with P defining l and m. In this case the definition of m implies that $m\leqslant n-l-2$ and $n-l-m\geqslant 2$. That means the number of missing edges is at least $2(l+1)$. If $2(l+1)\geqslant n-1$, which is equivalent to $l\geqslant (n-3)/2$, we are also done. We have proved the assertion if

$$(n-3)/2\leqslant n-\tfrac{1}{2}-\sqrt{3n-27/4}$$

which always is fulfilled for $n\geqslant 4$.

Case 2. K_{n-l}, the complete subgraph on $n-l$ vertices, is the minimal graph with P defining l and m. In this situation a minimal graph with P, namely K_{n-l}, has $\binom{r}{2}$ edges where $r=n-l$.

By (*) we can conclude $\binom{r}{2} = \frac{1}{2}r(r-1) \leqslant n-2$ and since r is an integer that

$$r \leqslant \lfloor \tfrac{1}{2} + \sqrt{2n - 15/4} \rfloor.$$

In this situation we have to count the number of missing edges more carefully. We investigate again the maximal graph G with $\bar{P}$ of Fig. 3. Since $m \geqslant 1$ we have at least $l+1 = n-r+1$ missing edges between K_{n-l-1} and the set of the other $l+1$ vertices. We like to prove the existence of at least $r-2$ more missing edges. These edges may miss between the two subgraphs or in the subgraph on $l+1$ vertices. Let z be the number of additional missing edges between the two subgraphs. If $z \geqslant r-2$ we are done. Otherwise, we have to prove the existence of at least $r-2-z$ missing edges on the $l+1$ vertices. We have at least $n-r+1-z$ vertices among these $l+1$ vertices where only one edge to K_{n-l-1} is missing. This set of vertices is called A. If v and w ($v, w \in A$) have the same missing neighbor in K_{n-l-1} the edge between v and w is missing. Otherwise v and w together with their $n-l-2$ common neighbors of K_{n-l-1} build a K_{n-l} and G has P. We call v and w equivalent if they have the same missing neighbor in $K_{n-l-1} = K_{r-1}$. Let $N_1, \ldots, N_{r-1}$ be the sizes of the equivalence classes. By our argument above all edges between two vertices of the same equivalence class are missing. Thus we have at least $\binom{N_1}{2} + \cdots + \binom{N_{r-1}}{2}$ additional missing edges. Here $\binom{0}{2} = \binom{1}{2} = 0$.

Altogether we have proved the existence of

$$n - r + 1 + z + \binom{N_1}{2} + \cdots + \binom{N_{r-1}}{2}$$

missing edges where $N_1 + \cdots + N_{r-1} \geqslant n-r+1-z$. If $N_1 + \cdots + N_{r-1}$ is given the sum of all $\binom{N_i}{2}$ takes its minimal value if $|N_i - N_j| \leqslant 1$ for all i, j. If the sum of all N_i is $2(r-2-z)+(z+1) = 2r-z-3$ the sum is minimal if $r-2-z$ of the N_i's equal 2 and the other $z+1$ N_i's equal 1. In this case the number of missing edges is $n-1$. Thus it is sufficient to prove that

$$n - r + 1 - z \geqslant 2r - z - 3 \qquad \text{or} \qquad r \leqslant \tfrac{1}{3}n + \tfrac{4}{3}.$$

We know already (see above) that $r \leqslant \lfloor \frac{1}{2} + \sqrt{2n - 15/4} \rfloor$. Thus we are done if $\frac{1}{2} + \sqrt{2n - 15/4} \leqslant \frac{1}{3}n + \frac{4}{3}$.

It is easy to see that the roots of this inequality are $n = 5$ and $n = 8$. Only the cases $n = 6$ and $n = 7$ are not covered by this inequality. But for these cases $\lfloor \frac{1}{2} + \sqrt{2n - 15/4} \rfloor = 3 \leqslant \frac{1}{3}n + \frac{4}{3}$. Altogether we have proved our theorem. ∎

5. The CREW–PRAM Complexity of Some Functions with Small Critical Complexity

For the classes B_n and M_n we have shown that the monotone address function is at least the nearly easiest function with respect to the critical complexity, perhaps even the easiest one. The graph property P: "No vertex is isolated" is the easiest one in MG_n and probably also in G_n with respect to the critical complexity. We show that for these two functions the $\log_b c(f)$-lower bound for the CREW–PRAM complexity is tight at least up to constant factors.

Proposition 4. *The graph property P: "No vertex is isolated" can be computed in time $O(\log n)$ by a CREW–PRAM with $O(n^2)$ processors.*

Proof. We describe the Boolean function f_P for the graph property P.

$$f_P(x_{1,2},..., x_{n-1,n}) = \bigwedge_{1 \leqslant i \leqslant n} \left(\bigvee_{1 \leqslant j \leqslant i-1} x_{j,i} \vee \bigvee_{i+1 \leqslant j \leqslant n} x_{i,j} \right).$$

It is easy to see that any associative and commutative operation on n variables can be computed by a CREW–PRAM in $O(\log n)$ steps. Here we compute in parallel for all vertices i whether they are isolated (the inner disjunctions). Afterwards, we compute f_P in $O(\log n)$ steps by the outer conjunction. ▌

Proposition 5. *The monotone address function g_m can be computed in time $O(\log \log m)$ by a CREW–PRAM with $O(m \log m)$ processors.*

Proof. It is equivalent to prove that the monotone address function on $n + \binom{n}{\lfloor n/2 \rfloor}$ variables can be computed in time $O(\log n)$ by a CREW–PRAM. Here the disjunction has $\Omega(2^n n^{-1/2})$ terms. Since the disjunction of k variables is k-critical a disjunction of $\Omega(2^n n^{-1/2})$ terms needs $\Omega(n)$ steps. Therefore we use another approach. In parallel we compute in time $O(\log n)$ all prime implicants $t_A(x, y) = \bigwedge_{i \in A} x_i \wedge y_{\alpha(A)}$ for all $A \subseteq \{1,..., n\}$, where $|A| = \lfloor n/2 \rfloor$ and we compute $s = x_1 + \cdots + x_n$. Let processor $p(A)$ hold $t_A(x, y)$ and processor p^* hold s. p^* writes 1 into the output cell if $s \geqslant \lfloor n/2 \rfloor + 1$ and 0 otherwise. All processors $p(A)$ read this cell. If $p(A)$ reads 1 or $t_A(x, y) = 0$ processor $p(A)$ does nothing. If $p(A)$ reads 0 and $t_A(x, y) = 1$ processor $p(A)$ writes 1 into the output cell. If p^* has written 0 there are at most $\lfloor n/2 \rfloor$ ones among the x variables and at most one processor $p(A)$ holds the result 1. Therefore our program meets the write restriction of CREW–PRAM's. ▌

Received: January 15, 1985; accepted: May 22, 1985

REFERENCES

COOK, S. A., AND DWORK, C. (1982), Bounds on the time for parallel RAM's to compute simple functions, *in* "Proceedings of the 14th Annu. ACM Sympos. Theory of Comput.," 231–233.

COOK, S. A., DWORK, C., AND REISCHUK, R. (*in press*), Upper and lower time bounds for parallel random access machines without simultaneous writes, *SIAM J. Comput.*

SIMON, H. U. (1983), A tight $\Omega(\log\log n)$-bound on the time for parallel RAM's to compute nondegenerate Boolean functions, *in* "Sympos. on Foundations of Comput. Theory," Lecture Notes in Computer Science, Vol. 158, pp. 439–444. Springer–Verlag, New York/Berlin.

TURÁN, G. (1984), The critical complexity of graph properties, *Inform. Process. Lett.* **18**, 151–153.